

FORM PTO-1390
(REV 10-96)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES
DESIGNATED/ELECTED OFFICE (DO/EO/US)
CONCERNING A FILING UNDER 35 U.S.C. 371

GDC-136

U.S. APPLICATION NO. (If known, see 37 CFR 1.5)

09/806783

INTERNATIONAL APPLICATION NO.

PCT/US99/22651

INTERNATIONAL FILING DATE

29 September 1999

PRIORITY DATE CLAIMED

2 October 1998

TITLE OF INVENTION

Network Management Information Processing

APPLICANT(S) FOR DO/EO/US

GYMER, David and BURDEN, Paul

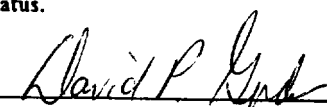
Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☐ This express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371(c)(2))
 - a. ☐ is transmitted herewith (required only if not transmitted by the International Bureau).
 - b. ☐ has been transmitted by the International Bureau.
 - c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☐ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
 - a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
 - b. ☐ have been transmitted by the International Bureau.
 - c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
 - d. ☐ have not been made and will not be made.
8. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:

11. ☐ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☒ A **FIRST** preliminary amendment.
☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
14. ☐ A substitute specification.
15. ☐ A change of power of attorney and/or address letter.
16. ☐ Other items or information:

JC08 Rec'd PCT/PTO 3 0 MAR 2001

US APPLICATION NO. 09/806783 INTERNATIONAL APPLICATION NO. PCT/US99/22651		ATTORNEY'S DOCKET NUMBER GDC-136					
17. <input checked="" type="checkbox"/> The following fees are submitted: BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) : Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO \$1000 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO \$860 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO \$710 International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) \$690 International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) \$100 ENTER APPROPRIATE BASIC FEE AMOUNT =		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: left;">CALCULATIONS</th> <th style="text-align: left;">PTO USE ONLY</th> </tr> <tr> <td style="height: 100px; vertical-align: bottom;"> \$ 100.00 </td> <td></td> </tr> </table>		CALCULATIONS	PTO USE ONLY	\$ 100.00	
CALCULATIONS	PTO USE ONLY						
\$ 100.00							
Surcharge of \$130.00 for furnishing the oath or declaration later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492(e)).		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">\$</td> <td style="width: 50%;"></td> </tr> </table>		\$			
\$							
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE				
Total claims	26 - 20 =	6	X \$18.00				
Independent claims	4 - 3 =	1	X \$80				
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ \$270				
TOTAL OF ABOVE CALCULATIONS =			\$ 288.00				
Reduction of 1/2 for filing by small entity, if applicable. A Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28).			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">\$</td> <td style="width: 50%;"></td> </tr> </table>	\$			
\$							
SUBTOTAL =			\$ 288.00				
Processing fee of \$130.00 for furnishing the English translation later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492(f)).			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">\$</td> <td style="width: 50%;"></td> </tr> </table>	\$			
\$							
TOTAL NATIONAL FEE =			\$ 288.00				
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). \$40.00 per property			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">\$</td> <td style="width: 50%;"></td> </tr> </table>	\$			
\$							
TOTAL FEES ENCLOSED =			\$ 288.00				
			Amount to be:				
			refunded				
			charged				
a. <input checked="" type="checkbox"/> A check in the amount of \$ <u>288.00</u> to cover the above fees is enclosed.							
b. <input type="checkbox"/> Please charge my Deposit Account No. _____ in the amount of \$ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.							
c. <input checked="" type="checkbox"/> The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. _____. A duplicate copy of this sheet is enclosed.							
07-1732							
NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.							
SEND ALL CORRESPONDENCE TO:							
David P. Gordon 65 Woods End Road Stamford, CT 06905		SIGNATURE:  David P. Gordon NAME 29,996 REGISTRATION NUMBER					

09/806783

JC08 Rec'd PCT/PTO 3 0 MAR 2001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: David Gymer et al.

INTERN'L APPL'N NO.: PCT/US99/22651

GROUP ART UNIT:

FILED:

EXAMINER:

FOR: Network Management Information
Processing

ATT'Y DOCKET NO.: GDC-0136

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Sir:

PRELIMINARY AMENDMENT

Please amend the claims of the application as indicated on the attached pages of marked-up claims.

PRELIMINARY REMARKS

The claims have been amended and new claims have been added to remove multiple dependencies. The claims have been indicated as being in acceptable form for allowance. Enclosed are replacement pages for pages 18-21 which now number 18-24. Claims have been renumbered and reorganized in view of added claims.

Respectfully submitted,



David P. Gordon
Reg. #29,996
Attorney for Applicant(s)

65 Woods End Road
Stamford, CT 06905
(203) 329-1160

MARKED UP CLAIMS FOR PCT/US99/22651

Claims:

1. A method of supplying data from a table in a device which is responsive to network management protocol commands, the method comprising receiving a Protocol Data Unit designated as a table block access request;

identifying the Protocol Data Unit as a table block access request;

obtaining an Object Identifier of a table to be read from the Protocol Data Unit;

obtaining an index to a row to be read from the table from the Protocol Data Unit;

determining the number of rows to be read based on information obtained from the Protocol Data Unit;

looking up information in the table based on the Object Identifier and the index to the row to be read;

composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read;

outputting the response packet.

2. A method according to Claim 1, wherein Object Identifiers are only included in the response packet if requested.

3. (amended) A method according to Claim 1 [or Claim 2], wherein if Object Identifiers for the rows are to be included in the response packet, a single Object Identifier is included for each row.

4. (new) A method according to Claim 2, wherein if Object Identifiers for the rows are to be included in the response packet, a single Object Identifier is included for each row.

[4]5. (amended) A method according to Claim 2 or [Claim 3] wherein abbreviated Object Identifiers are included in the response packet.

6. (new) A method according to Claim 3 wherein abbreviated Object Identifiers are included in the response packet.

[5]7. (amended) A method according to [any preceding] claim 1 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

8. (new) A method according to claim 4 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

9.(new) A method according to claim 6 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

[6]10. (amended) A method according to [any preceding] claim 1 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

11. (new) A method according to claim 4 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

12. (new) A method according to claim 6 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

13. (new) A method according to claim 7 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

[7]14. A method according to Claim 6, wherein the column identifier information is in the form of index information.

[8]15. A method, in a network management device which issues and accepts network management protocol Protocol Data Units, of obtaining data from a table in a remote device, preferably arranged to perform a method according to any preceding claim, the method comprising:

determining:- (a) an Object Identifier of a table in the remote device to be accessed;

(b) an index to the start of a block of rows from which data within the table is required;

(c) the number of rows to be accessed;

composing a Protocol Data Unit designated as a table block access request and including information representative of said determining;

outputting the Protocol Data Unit to the remote device; and

obtaining said data from a response Protocol Data Unit received from the remote device.

[9]16. A method according to Claim [8] 15 further comprising determining whether the received Protocol Data Unit contains all the data requested and, if not, composing a further request for data.

[10]17. A method according to Claim [8] 15 [or Claim 9] further comprising supplying the data to a management application.

18. (new) A method according to Claim 16 further comprising supplying the data to a management application.

[11]19. (amended) A method according to [any preceding] claim 1, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

20. (new) A method according to claim 4, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

21. (new) A method according to claim 6, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

22. (new) A method according to claim 12, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

[12]23. A network device comprising:

means for responding to Protocol Data Units received containing network management protocol commands;

means for identifying a received Protocol Data Unit designated as a table block access request;

means for indexing a portion of a stored table based on (a) an Object Identifier and (b) an index to a row to be read from the

table, obtained from the Protocol Data Unit;

means for determining the number of rows to be read based on information obtained from the Protocol Data Unit;

means for looking up information in the table based on the Object Identifier and the index to the row to be read;

means for composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read.

[13]24. (amended) A device according to Claim [12] 23, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

[14]25. A Protocol Data Unit comprising:

an identifier signifying that the Protocol Data Unit is a table block access request;

an Object Identifier of a table to be accessed;

an index to a row within the table to be accessed;

information identifying the number of rows to be accessed.

[15]26. (amended) A Protocol Data Unit according to Claim [12] 25 further comprising information identifying the number of columns in the table to be accessed and an identifier for each column.

2/PRTS

1

09/806783

JC08 Rec'd PCT/PTO 30 MAR 2001

NETWORK MANAGEMENT INFORMATION PROCESSING

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the processing of information for network management. More particularly, but not exclusively, the present invention relates to the processing of information contained in tables for controlling a network.

2. State of the Art

In a distributed network such as the Internet, it is necessary to store various parameters, including routing information, at distributed points across the network and to extract that information for overall management of the network. Since different devices in a network may be made by different manufacturers and be of different types, it is desirable for communication of this information to be substantially device independent.

The Simple Network Management Protocol (SNMP) together with associated Management Information Base (MIB) structures have been designed to achieve device-independent management of a network and are widely used across the Internet. Basic details of SNMP may be found in any of a number of texts on the subject, an example of which is The Simple Book (An Introduction to Management of TCP/IP-based internets) by Marshall T. Rose published by Prentice-Hall 1991, the entire disclosure of which is incorporated herein by reference.

SUMMARY OF THE INVENTION

The invention is particularly concerned with the manipulation of data in tables such as a Management Information Base (MIB). Details of the structure of a MIB may be found in chapter 4, pages 91-130 of The Simple Book, referenced above.

Entries within a MIB are associated with Object Identifiers (OIDs) which may be lengthy strings. The invention is particularly concerned with access to tables (such as a MIB) using network management protocols such as the Simple Network Management Protocol (SNMP), a discussion of which may be found in chapter 5, pages 131-186, of The Simple Book.

In order to extract information from a table such as a MIB, SNMP defines Protocol Data Units (PDUs) for exchanging messages and commands and provides a "Get" command and a "Get Next" command which allow information to be retrieved and a table to be traversed effectively. The "Get Next" operator is described on pages 140-142 of the Simple book. Whilst the "Get Next" operator is a powerful tool for traversing a table, it can be inefficient if blocks of data are to be accessed.

Version 2 of the Simple Network Management Protocol provides a "Get Bulk" operator which effectively performs repeated "Get Next" operations. This can lead to significant improvements in efficiency compared to multiple "Get Next" operations. This can result in a significant saving of Protocol Data Units (PDUs) which must be exchanged and also in the total number of bytes which must flow across the network.

However, pursuant to the invention, it has been appreciated that more efficient access to large tables may yet be possible, preferably in a manner not incompatible with existing SNMP architecture. Studies pursuant to the present invention have revealed that a significant amount of the data transferred may comprise Object Identifiers (OIDs). Pursuant to the invention, it has been appreciated that complete OIDs do not necessarily need to be transmitted in every case. It has also been found, pursuant to the invention, that certain operations such as the extraction of a relatively small portion of a relatively large table may be inefficient even when using the "Get Bulk" operation.

It is an aim of the invention to provide methods of extracting data from tables which are compatible with existing network management protocol (such as SNMP) interactions, but which may provide improved efficiency.

According to a first aspect, the invention provides a method of supplying data from a table in a device which is responsive to network management protocol, commands preferably Simple Network Management Protocol commands. The method preferably comprises eight steps:

receiving a Protocol Data Unit designated as a table block access request;

identifying the Protocol Data Unit as a table block access request;

obtaining an Object Identifier of a table to be read from the Protocol Data Unit;

obtaining an index to a row to be read from the table from the Protocol Data Unit;

determining the number of rows to be read based on information obtained from the Protocol Data Unit;

looking up information in the table based on the Object Identifier and the index to the row to be read;

composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read;

outputting the response packet.

By providing an Object Identifier for the table and an index to a row (preferably the start row), lengthy Object Identifiers need not be communicated for every row or every table entry. Furthermore, the method may allow immediate access to a given block of rows, for example in the middle of the table, even when the Object Identifiers of those rows are not known.

It will be appreciated that the Simple Network Management Protocol is reviewed and updated from time to time and modifications are proposed. In this specification, which term includes the claims, references to Simple Network Management

Protocol includes derivatives and modifications of the protocol current at the time of filing (whether including enhanced, reduced or alternative functionality); indeed, a modified version of the basic protocol incorporating table access as defined herein is intended to be encompassed by the term. Devices which are responsive to a subset or derivative of SNMP commands are intended to be encompassed by the invention.

Another advantage is that the Object Identifiers of the rows and objects within the table need not be communicated in the response packet; preferably Object Identifiers are only communicated in the response packet if specifically requested. Preferably, if Object Identifiers for the rows are requested, a single Object Identifier, preferably abbreviated, is communicated for each row. It is well-known that Object Identifiers are hierarchical, the Object Identifier of an item within a table comprising the Object Identifier of the table with suffixes dependent on the row and column within the table. By "abbreviated" is meant sufficient identification information from the suffixes, optionally pre-pended with a further portion of the complete Object Identifier or a dummy prefix, but not including the entire Object Identifier.

Preferably, information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested. This may facilitate determination by the requestor of the amount of information supplied and composition of a subsequent request for remaining information.

Preferably, the method includes selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit. This may allow data to be selectively extracted from multiple columns and multiple rows within a single operation. Most preferably, the column identifier information is in the form of index information. This avoids the need to communicate the

Object Identifier to each column, and allows specified columns to be accessed even when the Object Identifiers are not known.

In a second aspect, the invention provides a method, in a network management device which issues and accepts network management protocol, preferably Simple Network Management Protocol, Protocol Data Units, of obtaining data from a table in a remote device, preferably arranged to perform a method as defined above. The method preferably comprises six steps:

determining an Object Identifier of a table in the remote device to be accessed;

determining an index to the start of a block of rows from which data within the table is required;

determining the number of rows to be accessed;
composing a Protocol Data Unit designated as a table block access request and including information representative of on or more of said determining steps;

outputting the Protocol Data Unit to the remote device; and
obtaining said data from a response Protocol Data Unit received from the remote device.

Preferably, the method further comprises determining whether the received Protocol Data Unit contains all the information requested and, if not, composing a further request for information.

The method may further comprise supplying the information to a management application.

In a third aspect, the invention provides a network device comprising:

means for responding to Protocol Data Units received containing network management protocol, preferably Simple Network Management Protocol, commands;

means for identifying a received Protocol Data Unit designated as a table block access request;

means for indexing a portion of a stored table based on an Object Identifier and an index to a row to be read from the table from the Protocol Data Unit;

means for determining the number of rows to be read based on information obtained from the Protocol Data Unit;

means for looking up information in the table based on the Object Identifier and the index to the row to be read; and

means for composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read.

According to a fourth aspect, the invention provides a Protocol Data Unit comprising:

an identifier signifying that the Protocol Data Unit is a table block access request;

an Object Identifier of a table to be accessed;

an index to a row within the table to be accessed; and

information identifying the number of rows to be accessed.

The Protocol Data Unit preferably further comprises information identifying the number of columns in the table to be accessed and an identifier for each column.

It will be appreciated that the invention can be applied regardless of the information contained within the table to the access and provide a technical improvement in terms of more efficient data transfer and simplified access to large tables.

An embodiment of the invention will now be described, by way of example, with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a graph illustrating a comparison between the amount of data to be transferred when access a large table according to conventional methods and according to an embodiment of the invention;

Fig. 2 is a graph illustrating a comparison between the amount of Protocol Data Units to be transferred when access a large table according to conventional methods and according to an embodiment of the invention;

Fig. 3 is a graph illustrating a comparison between the amount of time taken for table retrieval when access a large table according to conventional methods and according to an embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment for use in an SNMP-compatible network device having a plurality of MIB tables stored therein will now be described. Details of conventional MIB tables and SNMP commands, together with details of Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules (BER) encoding are assumed to be well-known and will not be described in detail; reference should be made to The Simple Book, together with references 40-53 in the bibliography thereon, or to any of the relevant standards, all of which are incorporated herein by reference.

By way of background summary information, basic formats of an SNMP message, a generic PDU, a request PDU, a Get PDU and a Get Next PDU will be set out, in ASN.1 syntax.

Firstly, a basic message format:-

-- top-level message

Message ::=

SEQUENCE {

version -- version-1 for this RFC

INTEGER {

version-1(0)

},

community -- community name

OCTET STRING,


```

        data          -- e.g., PDUs if trivial
        ANY           -- authentication is being used
    }

```

Next, the format of a Protocol Data Unit:-

```
-- protocol data units
```

```

PDUs ::=
    CHOICE {
        get-request
            GetRequest-PDU,

        get-next-request
            GetNextRequest-PDU,

        get-response
            GetResponse-PDU,

        set-request
            SetRequest-PDU,

        trap
            Trap-PDU
    }

```

```
-- the individual PDUs and commonly used
```

```
-- data types will be defined later
```

```
END
```

The basic format of a request PDU will now be set out:-

```
-- request/response information
```

```

RequestID ::=
    INTEGER

```

```

ErrorStatus ::=
    INTEGER {

```

9

```

noError(0),
tooBig(1),
noSuchName(2),
badValue(3),
readOnly(4)
genErr(5)

```

```

}

```

```

ErrorIndex ::=
    INTEGER

```

```

-- variable bindings

```

```

VarBind ::=
    SEQUENCE {
        name
            ObjectName,

        value
            ObjectSyntax
    }

```

```

VarBindList ::=
    SEQUENCE OF
        VarBind

```

The format of a standard "Get" PDU is:-

```

GetRequest-PDU ::=
    [0]
        IMPLICIT SEQUENCE {
            request-id
                RequestID,

            error-status          -- always 0
                ErrorStatus,

            error-index          -- always 0
                ErrorIndex,

```

```

variable-bindings
  VarBindList
}

```

The format of a "Get Next" PDU is:-

```

GetNextRequest-PDU ::=
  [1]
    IMPLICIT SEQUENCE {
      request-id
        RequestID,

      error-status      -- always 0
        ErrorStatus,

      error-index      -- always 0
        ErrorIndex,

      variable-bindings
        VarBindList
    }

```

Further details of the components of the entities defined above and other background information may be found by reference to RFC 1157 or other standard texts.

According to this embodiment, we propose a modified PDU which we designate a Get Table Row message. This is defined below using the ASN.1 syntax:-

```

GetTableRow-PDU ::=
  SEQUENCE {
    request-id
      INTEGER,

    error-status
      INTEGER {
        noError(0),
        tooBig(1),
        noSuchName(2),
        badValue(3),
        readOnly(4),

```

```

                                genErr(5)
                                },

                                error-index
                                INTEGER,
                                snmpp-version
                                INTEGER {
                                    version1 (1)           -- First implementation
                                },

                                table-name
                                OBJECT-IDENTIFIER,         -- OID of the table being
retrieved
                                start-index
                                INTEGER,                   -- starting row index for
retrieval
                                max-rows
                                -- maximum no. of rows to be
retrieved
                                INTEGER,                   -- -1 indicates "get all rows"
                                table-size
                                INTEGER,                   -- No. of rows in table

                                instances-included
                                INTEGER {
                                    no(0),                  -- Row instances not encoded
                                    yes(1)                  -- Row instances are encoded
                                },

                                column-total
                                INTEGER,                   -- No. of columns to be retrieved

                                column1
                                INTEGER,                   -- column id for first column
                                column2
                                INTEGER,                   -- column id for second column
                                ...
                                columnN
                                INTEGER,                   -- column id for Nth column

                                variable-bindings
                                varBindList
                                }
VarBind ::=
    SEQUENCE {
        row-instance1
            OBJECT-IDENTIFIER,         -- optional instance OID for row
        value
            objectSyntax               -- value for this row/column entry
    }

```

This command is intended to allow a management application to retrieve arbitrary rows from a table without having to issue repeated GetNext commands to get to the correct rows. For optimum efficiency and flexibility, it is found to be highly

desirable that the command can access arbitrary columns, and not just complete rows.

An explanation of the fields in a GetTableRows request PDU as would be sent from a management application follows:-

- request-id
The unique request id for this PDU
- snmpp-version
Indicates the revision level of the SNMPP PDU (should always be set to 1).
- table-name
The OBJECT IDENTIFIER representing the table to be retrieved. For example, the interfaces table in rfc1213 would have a table name of 1.3.6.1.2.1.2.2
- start-index
Identifies the first row index to be retrieved from the table. This represents essentially the row number in that table (starting 0). So, to start retrieving from the first row, start-index would be set to 0. To retrieve from the 25th row, start-index would be set to 24, etc.
- max-rows
Represents the maximum number of rows to be retrieved (if possible). If all rows from the start-index to end of table are required, this should be set to -1.
- column-total
Represents the total number of columns to be retrieved from the table (the column ids are encoded immediately after this object in the PDU).
- column-id
A column id is encoded for each of the columns requested. So, for example, if five columns had been requested, then five consecutive INTEGERS would be encoded representing the respective column ids. The id represents the conceptual column number for that table (starting 1). So, for example, consider the ifTable of rfc1213, the column-id for ifOperStatus

would be 8, since this is the eighth conceptual column in the table.

The request PDU will contain an empty varbind list (since all the information above is sufficient to identify what we are requesting).

Note: All the other objects exist in the request PDU, but will have their default values set.

To implement this embodiment, the (modified) SNMP agent of the network device must process an incoming GetTableRows request and package the response message to send back to the requestor. The agent should attempt to include all the requested rows into the response PDU, but due to the restrictions of message size, this may not be possible. In these cases, it should send back as many rows as it can, updating the associated fields to identify precisely the rows it has returned (this is so that the requestor can send another GetTableRows request message amended to retrieve the remaining rows).

A GetTableRows response PDU should be sent to the management application with the following fields set:-

- request-id
The unique request id for this PDU.
- snmpp-version
Indicates the revision level of the SNMPP PDU (should always be set to 1)
- table-name
The OBJECT IDENTIFIER representing the table to be retrieved. For example, the interfaces table in rfc1213 would have a table name of 1.3.6.1.2.1.2.2. This must match the request PDU.
- start-index
Identifies the first row index to be retrieved from the table. This represents essentially the row number in that table (starting 0). So, to start retrieving

from the first row, start-index would be set to 0. To retrieve from the 25th row, start-index would be set to 24, etc. This must match the request PDU.

- max-rows

This will be set to the actual number of rows included in this response PDU.

- table-size

Stores the actual size of the table requested (i.e. how many rows exist in the table at that point in time).

- instances-included

set to no(0) if the row instances have not been encoded in the varbinds representing the first column requested, otherwise set to yes(1) if they have.

- column-total

Represents the total number of columns retrieved from the table (the column ids are encoded immediately after this object in the PDU). This must match the request PDU.

- column-id

A column id is encoded for each of the columns requested. So, for example, then five consecutive INTEGERS would be encoded representing the respective column ids. The id represents the conceptual column number for that table (starting 1). So, for example, consider the ifTable of rfc1213, the column-id for ifOperStatus would be 8, since this is the eighth conceptual column in the table. Each of these column-ids must match the request PDU.

- varbind list

A list of varbinds must be encoded which represent the data contained in the rows returned. The order of the varbind list is on a per-row basis. So, for example, if five columns had been requested, the first five varbinds would constitute the values for the first row returned, where varbind1 represents the data for column1, varbind2 contains the data for column2 and so

on. In most cases, the name of the varbind is not encoded (see the later section on varbind encoding).

The SNMP GetTableRows message is encoded with a message type of 0xAF, which corresponds to:-

ASN_CONTEXT 1 ASN_CONSTRUCTOR 1 0xf

A variable binding list returned in a GetTableRows response message will contain each of the values within the table encoded as usual varbind objects. The varbind list must always contain enough variables encoded in the varbind list will be multiples of column-total.

The variable binding for each element in a row will be encoded in order of column-ids requested. The object-name of a varbind will only be encoded if the following two criteria are met:-

1. The instances-included variable is set to yes(1)
2. The varbind being encoded represents the first column-id of a row.

If the object name is encoded, it will represent the instance oid identifying that row (starting with 0.0, because the first two subids must each be encoded in a single octet according to SNMP).

This is best explained by example, so consider the ifTable and the TableRows request message has requested two columns, namely ifAdminStatus (1.3.6.1.2.1.2.2.1.7) and ifOperStatus (1.3.6.1.2.1.2.2.1.8).

The column-ids will be encoded as two INTEGERS, namely 7 and 8.

Supposing the response message was returning 3 rows (for ifIndex 1,2 and 3). The varbind list will be encoded as follows:-

Varbind	Object Name (row-Instance)	Value
1	0.0.1	up(1)
2	Not Encoded	up(1)
3	0.0.2	up(1)
4	Not Encoded	down(2)
5	0.0.3	testing (3)
6	Not Encoded	Down(2)

The above varbinds would represent the following three rows in the **ifTable**:-

ifIndex	ifAdminStatus	ifOperstatus
1	up(1)	up(1)
2	up(2)	down(2)
3	testing(3)	down(2)

The following pseudo-code outlines the basic steps to be performed to implement the embodiment (some of which will co-exist with other steps which are part of a conventional SNMP agent) :-

- Receive PDU
- [Other SNMP processing]
- Check whether PDU designated "GetTableRows"
- If not so designated, skip to Continued Processing
- If so designated:-
 - Obtain OID of table to be read from table-name

- Obtain index to first row to read from start-index .
- Obtain number of rows to read from max-rows
- Obtain indices to columns to be read column1..N
- Check whether encoded row ids requested in instances-included
- Look up information in specified table using indices
- Compose response packet including:-
 - * Information read from table in varbinds
 - * Number of rows actually read in max-rows
 - * Row ids if specified in varbinds for first column
- Output response packet

[Continued Processing]

It will be appreciated that the ordering of information is not critical and can be changed, as can all labels used both for entities with the PDU and the PDU designation (the label GetTableRows being used here as a suitable label to designate a table block access request). The information contained in the PDU may be replaced by other combinations of information which achieve the same function (for example, the last row may be supplied in place of the first row, and the indexing performed in reverse). Not all functions need be included.

Each feature described above may be provided independently, unless otherwise stated.

Claims:

1. A method of supplying data from a table in a device which is responsive to network management protocol commands, the method comprising receiving a Protocol Data Unit designated as a table block access request;

identifying the Protocol Data Unit as a table block access request;

obtaining an Object Identifier of a table to be read from the Protocol Data Unit;

obtaining an index to a row to be read from the table from the Protocol Data Unit;

determining the number of rows to be read based on information obtained from the Protocol Data Unit;

looking up information in the table based on the Object Identifier and the index to the row to be read;

composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read;

outputting the response packet.

2. A method according to Claim 1, wherein Object Identifiers are only included in the response packet if requested.

3. A method according to Claim 1, wherein if Object Identifiers for the rows are to be included in the response packet, a single Object Identifier is included for each row.

4. A method according to Claim 2, wherein if Object Identifiers for the rows are to be included in the response packet, a single Object Identifier is included for each row.

5. A method according to Claim 2 wherein abbreviated Object Identifiers are included in the response packet.

6. A method according to Claim 3 wherein abbreviated Object Identifiers are included in the response packet.

7. A method according to claim 1 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

8. A method according to claim 4 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

9. A method according to claim 6 wherein information representative of the number of rows actually included in the response packet is included in the response packet, at least when the number of rows supplied differs from the number of rows requested.

10. A method according to claim 1 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

11. A method according to claim 4 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

12. A method according to claim 6 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

13. A method according to claim 7 including selecting one or more columns from which data is to be included based on column identifier information within the received Protocol Data Unit.

14. A method according to Claim 6, wherein the column identifier information is in the form of index information.

15. A method, in a network management device which issues and accepts network management protocol Protocol Data Units, of obtaining data from a table in a remote device, preferably arranged to perform a method according to any preceding claim, the method comprising:

determining:- (a) an Object Identifier of a table in the remote device to be accessed;

(b) an index to the start of a block of rows from which data within the table is required;

(c) the number of rows to be accessed;

composing a Protocol Data Unit designated as a table block access request and including information representative of said determining;

outputting the Protocol Data Unit to the remote device; and

obtaining said data from a response Protocol Data Unit received from the remote device.

16. A method according to Claim 15 further comprising determining whether the received Protocol Data Unit contains all the data requested and, if not, composing a further request for data.

17. A method according to Claim 15 further comprising supplying the data to a management application.

18. A method according to Claim 16 further comprising supplying the data to a management application.

19. A method according to claim 1, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

20. A method according to claim 4, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

21. A method according to claim 6, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

22. A method according to claim 12, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

23. A network device comprising:

means for responding to Protocol Data Units received containing network management protocol commands;

means for identifying a received Protocol Data Unit designated as a table block access request;

means for indexing a portion of a stored table based on (a) an Object Identifier and (b) an index to a row to be read from the table, obtained from the Protocol Data Unit;

means for determining the number of rows to be read based on

information obtained from the Protocol Data Unit;

means for looking up information in the table based on the Object Identifier and the index to the row to be read;

means for composing a response Protocol Data Unit containing information read from the table for a plurality of rows based on the number of rows to be read.

24. A device according to Claim 23, wherein the network management protocol is Simple Network Management Protocol, or a derivative or modification thereof.

25. A Protocol Data Unit comprising:

an identifier signifying that the Protocol Data Unit is a table block access request;

an Object Identifier of a table to be accessed;

an index to a row within the table to be accessed;

information identifying the number of rows to be accessed.

26. A Protocol Data Unit according to Claim 25 further comprising information identifying the number of columns in the table to be accessed and an identifier for each column.

ABSTRACT OF THE DISCLOSURE

A method for supplying data from a table in a device responsive to network management protocol commands includes receiving a Protocol Data Unit (PDU) designated as a table block access request (TBAR), identifying the PDU as a TBAR, obtaining an Object Identifier (OI) of a table to be read from the PDU, obtaining an index to a row to be read from the table from the PDU, determining the number of rows to be read based on information obtained from the PDU, looking up information in the table based on the OI and the index, composing a response PDU containing information read from the table for multiple rows based on the number of rows to be read, and outputting a response packet (RP). Optionally, OIs are only included in the RP if requested, and abbreviated OIs are included in the RP. Network devices implementing the method are also provided.

09/806783

1/2

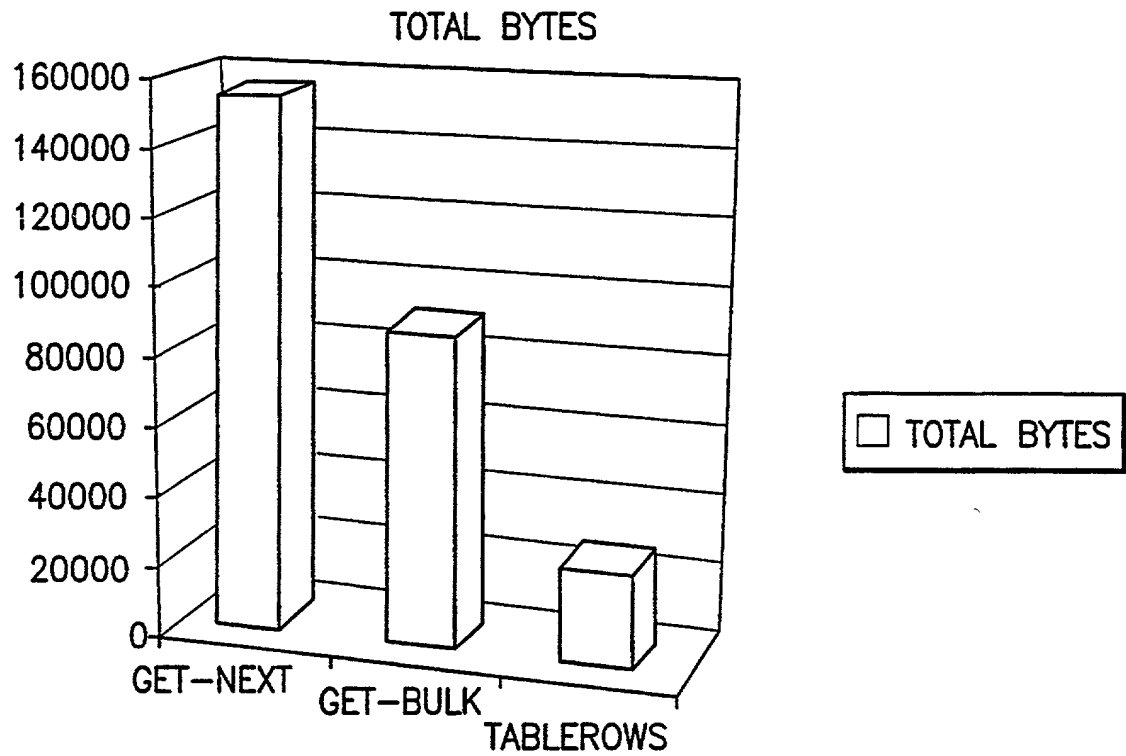


FIG.1

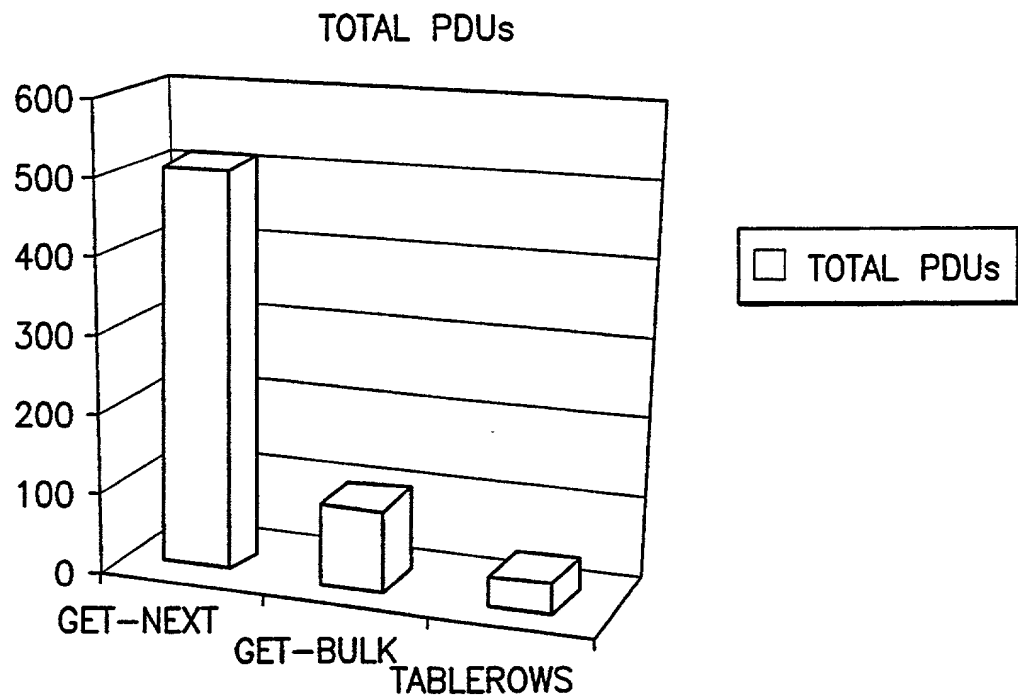


FIG.2

09/806783

2/2

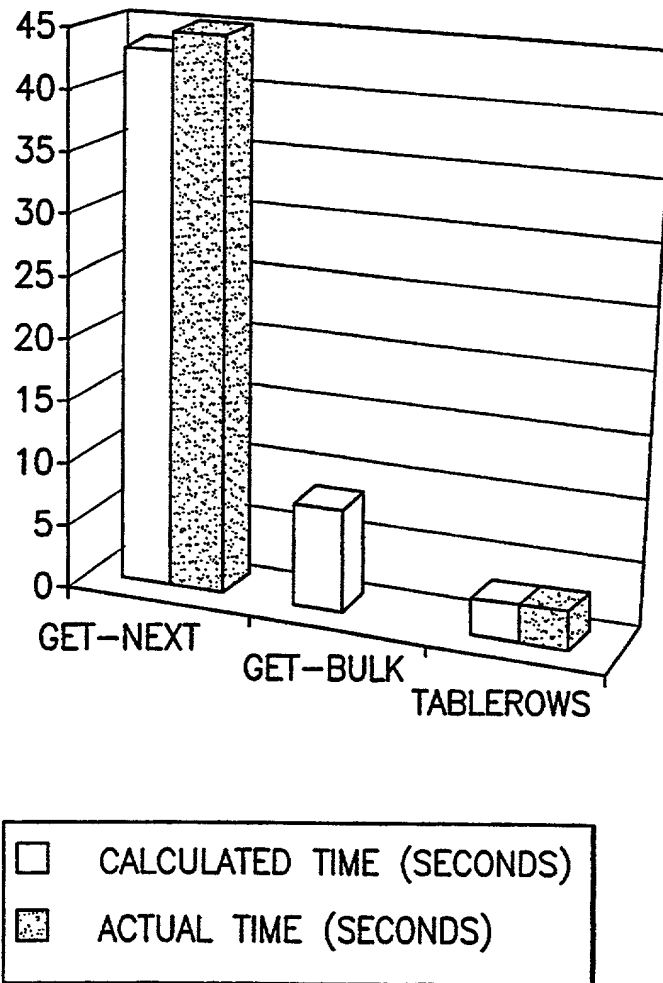


FIG.3

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name, and

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed for and for which a patent is sought on the invention entitled:

NETWORK MANAGEMENT INFORMATION PROCESSING,

the specification of which

☐ is attached hereto.

☐ was filed on _____

as application Serial Number _____

and was amended on (if applicable) _____

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)**Priority Claimed**

Serial No.	Country	Filing Date (D/M/YR)	Priority Claimed?
1. PCT/US99/22651	US	29/09/99	<input checked="" type="checkbox"/> YES <input type="checkbox"/> NO
2.			<input type="checkbox"/> YES <input type="checkbox"/> NO

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Ser. No) (Filing Date) (Status-Patented, pending, abandoned)

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

David P. Gordon (29,996)
David S. Jacobson (39,235)
Thomas A. Gallagher (31,358)

3

Address all telephone calls to: David P. Gordon at (203) 329-1160
Address all correspondence to: David P. Gordon, Esq.
65 Woods End Road
Stamford, Connecticut 06905
U.S.A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

SOLE OR FIRST INVENTOR

Signature [Signature] Date 26/March/2001

Full Name: David Gymer

COBX

Residence: 5 Randolph Close, Maldon, Essex CM9 6XP, United Kingdom

Citizenship: UK

P.O. Address: Same as address

SECOND JOINT INVENTOR

Signature _____ Date _____

Full Name: Paul Burden

Residence: 55 Cricketfield Grove, Leigh-on-Sea, Essex CC9 3EJ, United Kingdom

Citizenship: UK

P.O. Address: Same as address

DECLARATION FOR PATENT APPLICATION AND POWER OF ATTORNEY

As below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name, and

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed for and for which a patent is sought on the invention entitled:

NETWORK MANAGEMENT INFORMATION PROCESSING,

the specification of which

☐ is attached hereto.

☐ was filed on _____

as application Serial Number _____

and was amended on (if applicable) _____

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)**Priority Claimed**

Serial No.	Country	Filing Date (D/M/YR)	Priority Claimed?
1. PCT/US99/22651	US	29/09/99	[X] YES [] NO
2.			[] YES [] NO

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Ser. No) (Filing Date) (Status-Patented, pending, abandoned)

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

David P. Gordon (29,996)
David S. Jacobson (39,235)
Thomas A. Gallagher (31,358)

Address all telephone calls to: David P. Gordon at (203) 329-1160
Address all correspondence to: David P. Gordon, Esq.
65 Woods End Road
Stamford, Connecticut 06905
U.S.A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

SOLE OR FIRST INVENTOR

Signature _____ Date _____

Full Name: David Gymer

Residence: 5 Randolph Close, Maldon, Essex CM9 6XP, United Kingdom

Citizenship: UK

P.O. Address: Same as address

SECOND JOINT INVENTOR

Signature Paul Burden Date 18TH APRIL 2001

Full Name: Paul Burden

Residence: 55 Cricketfield Grove, Leigh-on-Sea, Essex CC9 3EJ, United Kingdom

Citizenship: UK

P.O. Address: Same as address